Design Automation for Design Synthesis

Saroj Bardewa, Aanchal Chobisa, Bhaskar Venkatramaiah

saroj@pdx.edu, aanchal@pdx.edu, vbhaskar@pdx.edu,

Abstract -- The importance of design automation in design synthesis is quite evident. Given the complexity of design synthesis, design automation simplifies the synthesis process by accelerating the design process as well as making design specific information readily available. Getting the right information in a short period of time further aids in design debugging, making appropriate design decisions, thereby expediting the design process and efficiency. In this report, we present two design independent automation scripts that meet those preceding objectives. The first script allows a designer to quickly get the information of cells in a path and total number of cells and displays them in the form of a histogram. The second scripts provides timing margins of cells, nets and I/O ports. Both the scripts are written in TCL and well tested and verified on Design Compiler.

I. INTRODUCTION

Design automation is widely used in industries to make a design synthesis flow faster as well as to facilitate more efficient design implementations. Not only do design implementations go smoother with design automation techniques, but also they provide valuable information relating to components such as cells, nets, pins, and ports that enable achieving a quality design within a given power, performance and area constraints. Scripting can help to automate the task of data collection for analysis as well. As a part of our final project, we were required to create two generic scripts for specific automation requirements and provide their outputs.

We designed two scripts that employed standard design automation techniques to extract key information relating to design components and timing from a synthesized design. As far as where our design automation endeavor would fit in the overall design synthesis is concerned, our work can directly assist backend designs in a design environment.

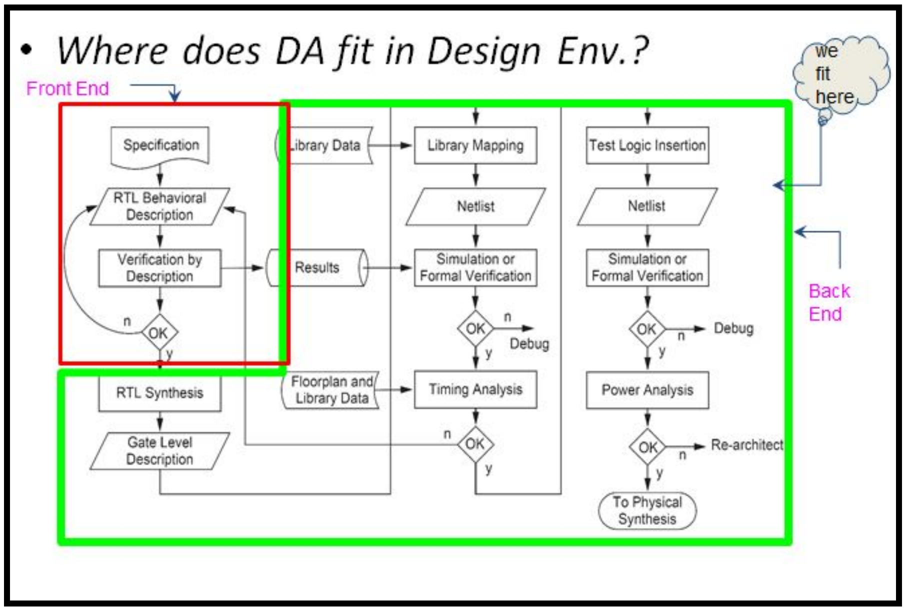
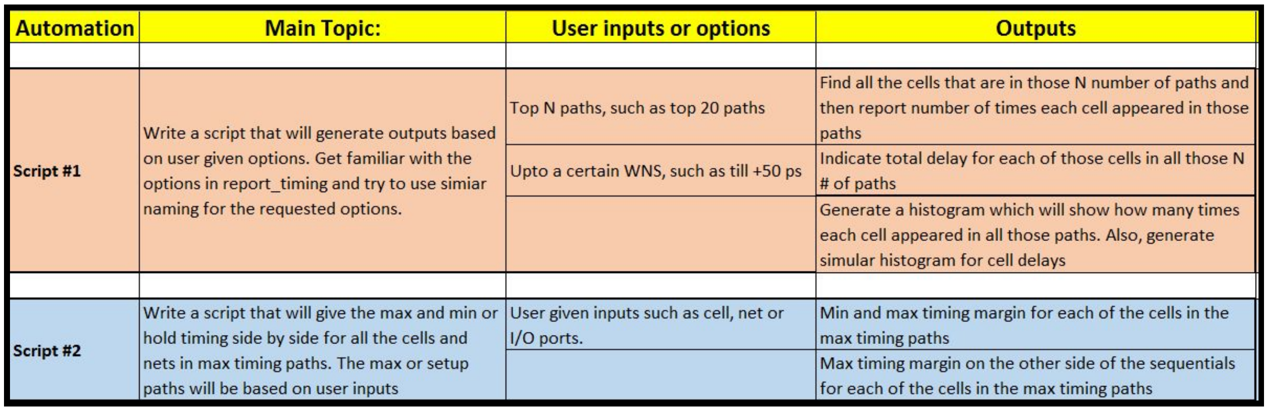


Figure 1: Design Automation in Design Environment



Automation	Main Topic:	User inputs or options	Outputs
Script #1	Write a script that will generate outputs based on user given options. Get familiar with the options in report_timing and try to use simiar naming for the requested options.	Top N paths, such as top 20 paths	Find all the cells that are in those N number of paths and then report number of times each cell appeared in those paths
		Upto a certain WNS, such as till +50 ps	Indicate total delay for each of those cells in all those N # of paths
			Generate a histogram which will show how many times each cell appeared in all those paths. Also, generate simular histogram for cell delays
Script #2	Write a script that will give the max and min or hold timing side by side for all the cells and nets in max timing paths. The max or setup paths will be based on user inputs	User given inputs such as cell, net or I/O ports.	Min and max timing margin for each of the cells in the max timing paths
			Max timing margin on the other side of the sequentials for each of the cells in the max timing paths

Figure 2: Table showing script #1 and scrip #2 specifications

The backend designers who work with logical and physical synthesis can greatly benefit from our designs. As shown in figure 1, our project designs can help to facilitate the backend synthesis process. The two TCL scripts we wrote meet our specific project needs and verified that their implementation facilitated to extract information in a backend design more quickly.

II. PROJECT DESCRIPTION

There were two primary requirements we strove to fulfil in our design that can directly leverage backend designs. Those needs were provided as part of project specifications. As tabulated in figure 2, there were two design goals to achieve as formulated under script #1 and script #2.

1) Script 1

The script 1 required us to write a script to generate cells on the top **N** paths, or cells on the path with the value till the user specified worst negative lack less. More precisely, the input to the script could be either top N path, Worst Negative Slack(WNS) or both and the output to be reported. The output of the script was to consist of all the cells in those N number of paths, the total number of times each cell appeared in those paths, the total delay of each cell in those paths, and histograms for total cell appearance count and cell delays.

2) Script 2

For the second script, the problem asked to give the minimum and maximum timing margin for each cell in the maximum timing path. Inputs could be either cells, nets or I/O ports. The output of the script was the minimum and the maximum timing margin of each cells in the maximum timing paths, as well as the maximum timing margin on the other side of the sequentials.

III. DESIGN IMPLEMENTATION

As the presented problems specified, we have written two two scripts that meet the needs of the project. We take an algorithmic approach to design the scripts and test them with a simplified design.

1) SCRIPT I

To meet the requirement of script 1, the following algorithm was designed:

```
1. Step 1: Get the input from user as a command argument
    1.1. If the user provides -help show the help information
    1.2. If the user provides only N paths
        1.2.1. Display report_timing  for the maximum N paths
        1.2.2. Get all paths that are less than or equal to N paths
        1.2.3. Create an empty array to store cells and their counts
        1.2.4. For each path get the starting point location
            1.2.4.1. For each point along the path
                1.2.4.1.1. If the array is empty store the cell on the array and continue
                           to move along the path
                1.2.4.1.2. Else, for each element in the array, compare if the cell is
                           already present in the array
                    1.2.4.1.2.1. If the cell is already present in the array, increment
                                 its count
                    1.2.4.1.2.2. Else, add the new cell to the member array, and
                                 initialize its count to 1.
                1.2.4.1.3. Repeat the process for all cells in the path
            1.2.4.2. Repeat this process for all paths in the N path.
    1.3. Show cell counts to display all elements of the array
    1.4. Show histogram of the cells to display all elements of the array
    1.5. Else if,  the user provides only WNS value
        1.5.1. Repeat the process 1.2  but for only a single path with slack less than
               WNS
        1.5.2. Repeat 1.3 and 1.4
    1.6. Else if , the user provides both , N paths and WNS,
        1.6.1. Repeat the process 1.2  but limit the report_timing to N paths and slack
               than WNS
        1.6.2. Repeat 1.3 and 1.4
    1.7. Else, show the input report information
```

In addition to implementing the above algorithm, for script 1, we use the application provided *report_timing* information to display the delay information of cell along each path. We also build our own histogram design technique.Please see appendix-A for code.

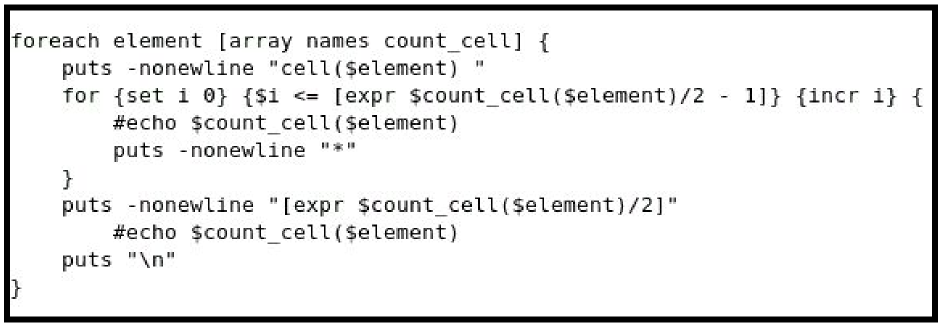
- *report_timing :*
  This command provides a report of timing information for the current design. The

command options allow us to specify the the amount of information reported. For our purpose, we customize the report timing specific to **N** paths and slack less than **WNS**.

- *get_attribute :* We also extensively use, *get_attribute* command to get the name, and start-end point of paths and cells.

- *Simple histogram :*

  To draw a simple histogram on a console, e first fill an array in the form {cell_a, count, cell_b, count, cell_c, count....} . Then, for every cell of the array, we display "*" , count number of times which signifies the number of times the cell appears in the design. The following shows the tcl-based implementation of the design:



```
foreach element [array names count_cell] {
    puts -nonewline "cell($element) "
    for {set i 0} {$i <= [expr $count_cell($element)/2 - 1]} {incr i} {
        #echo $count_cell($element)
        puts -nonewline "*"
    }
    puts -nonewline "[expr $count_cell($element)/2]"
        #echo $count_cell($element)
    puts "\n"
}
```

Figure 3: Showing tcl-based implementation of a simple histogram

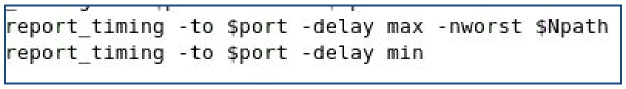
There were issues drawing a histogram with cell delays. Primarily, we were not able to obtain each cell delay from *get_attribute* command. Hence, the algorithm to draw histogram with cell delays is not included in this report.

2) Script II

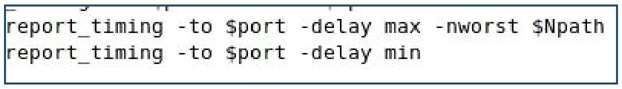
Script 2 has following algorithm:

```
1.  Get the user input
    1.1.  If the input is -help, display the help information
    1.2.  If the input command to display is -from_port {PortA}
          1.2.1.  Display max and min timing margin from {Port A} for each of the cells in
                  the maximum timing margin
          1.2.2.  Display max_rise and max_fall timing for each of the cells on the other
                  side of sequentials
    1.3.  If the input command to display is -to_port {PortA}
          1.3.1.  Repeat steps 1.2.1 and 1.2.2 for all the port to {Port A}
    1.4.  If the display command is to display all -cells
          1.4.1.  Repeat steps 1.2.1 and 1.2.2 for all cells
    1.5.  If the display command is to display all -nets
          1.5.1.  Repeat steps 1.2.1 and 1.2.2 for all nets
```

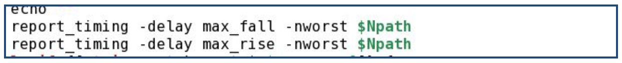
The implementation of the algorithm exclusively uses *report_timing* and changes the parameters according to the user input. Please see Appendix-B for more details.



```
report_timing -to $port -delay max -nworst $Npath
report_timing -to $port -delay min
```



```
report_timing -to $port -delay max -nworst $Npath
report_timing -to $port -delay min
```



```
echo
report_timing -delay max_fall -nworst $Npath
report_timing -delay max_rise -nworst $Npath
```

Figure 4: Showing example of six variations in report timing commands used to generate output

IV. TESTS AND RESULTS

To test both of our scripts, we use a simpler design as shown in figure 5. This simpler desing allowed us to manually verify that the results generated through our results is accurate. We reference this design as *SimpleAnd* design in our RTL module. Rather than any functional usefulness, this design circuit is merely chosen for simplification of our test and verification.

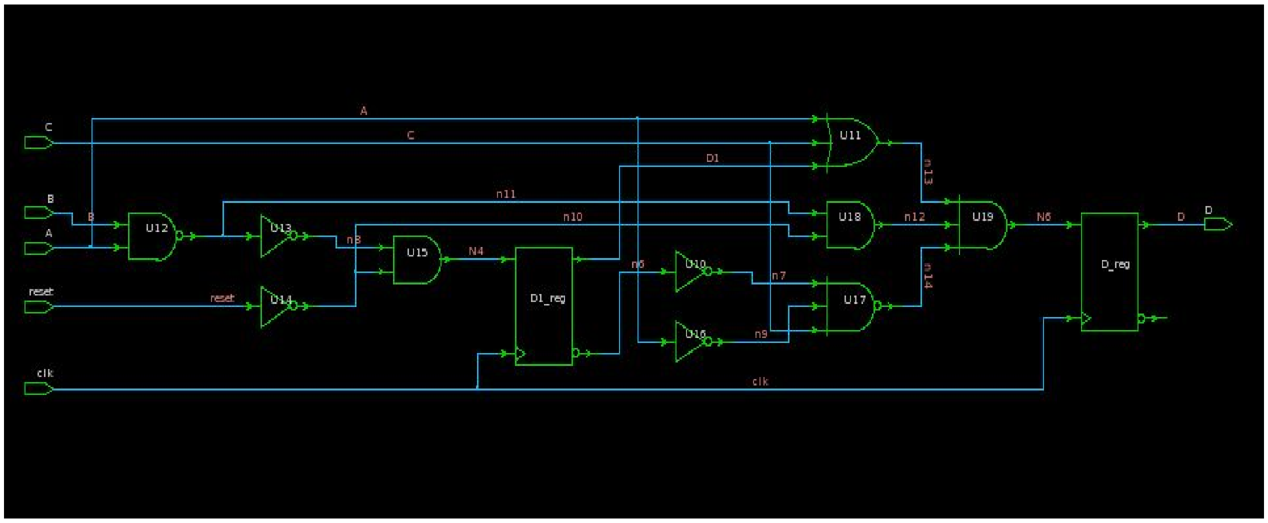


Figure 5: Simple circuit used for testing scripts

Next, we describe the outputs of our scripts:

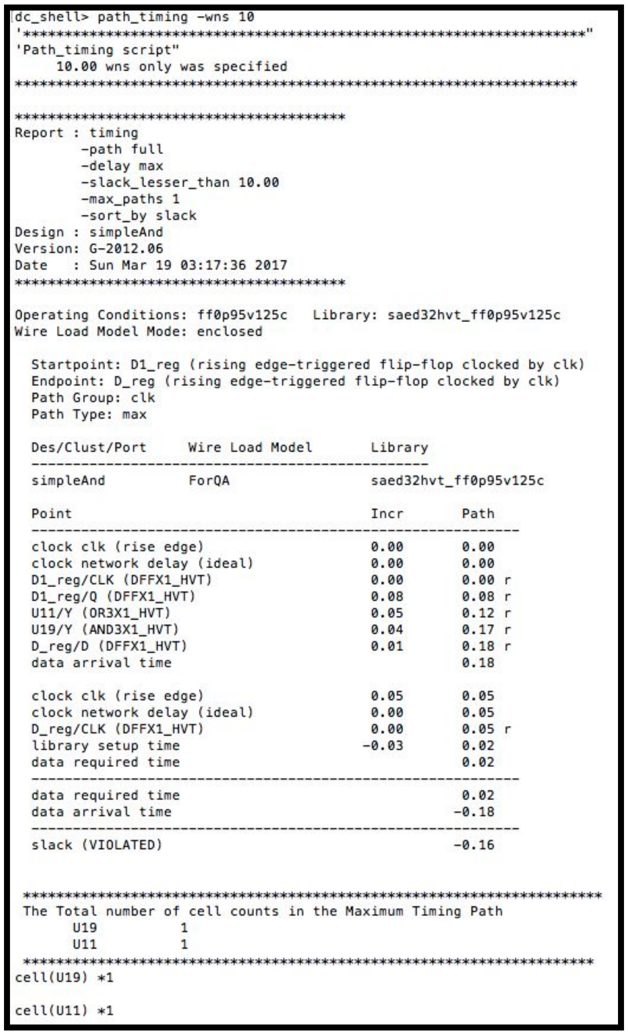
A) Script I outputs:

There are three possible user input combinations script 1 : path only, wns only or both path and wns. We obtain following results for each of the combinations in our circuit:

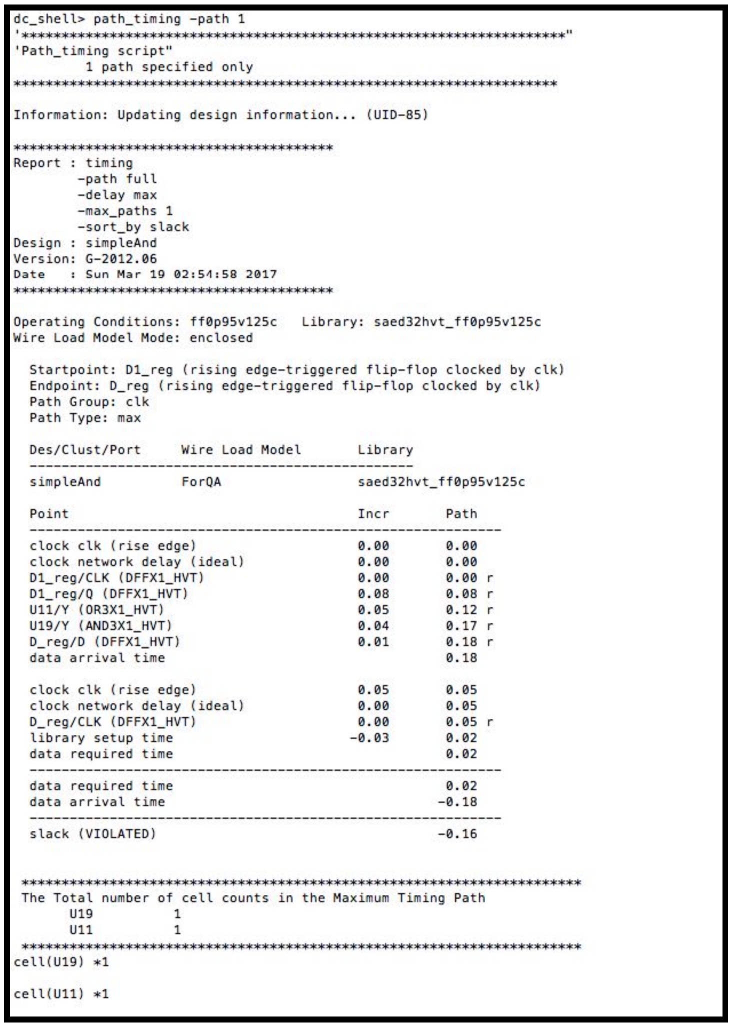
1. *Path and WNS Only :* Figure 5 shows results for inputs *path_timing -wns* 10 (left) and *path_timing -path 1 (right)* . It shows all the cell delay information for all cell paths, along with combinational cell counts and their respective simple histogram.

   Likewise, figure 6 shows an example of a simple histogram that is generated based on a given cell count in a design. As shown, the number of asterisk indicates the number of times the cells are repeated.

2. *Both WNS (X) and Path(N) specified* : When both WNS and Path are specified, the script generates all N paths specified that have slack less than X. Figure 7 shows the results for command : *path_timing -path 2 -wns 3* . The histogram shows the number of times the combinational blocks appeared in the 2 worst paths.

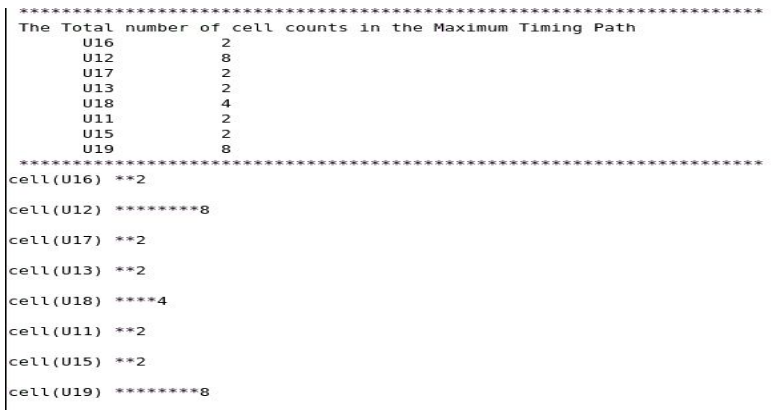


```
dc_shell> path_timing -wns 10
'*********************************************************************"
'Path_timing script"
     10.00 wns only was specified
*********************************************************************

****************************************
Report : timing
        -path full
        -delay max
        -slack_lesser_than 10.00
        -max_paths 1
        -sort_by slack
Design : simpleAnd
Version: G-2012.06
Date   : Sun Mar 19 03:17:36 2017
****************************************

Operating Conditions: ff0p95v125c   Library: saed32hvt_ff0p95v125c
Wire Load Model Mode: enclosed

  Startpoint: D1_reg (rising edge-triggered flip-flop clocked by clk)
  Endpoint: D_reg (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  simpleAnd          ForQA                 saed32hvt_ff0p95v125c

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  D1_reg/CLK (DFFX1_HVT)                   0.00       0.00 r
  D1_reg/Q (DFFX1_HVT)                     0.08       0.08 r
  U11/Y (OR3X1_HVT)                        0.05       0.12 r
  U19/Y (AND3X1_HVT)                       0.04       0.17 r
  D_reg/D (DFFX1_HVT)                      0.01       0.18 r
  data arrival time                                   0.18

  clock clk (rise edge)                    0.05       0.05
  clock network delay (ideal)              0.00       0.05
  D_reg/CLK (DFFX1_HVT)                    0.00       0.05 r
  library setup time                      -0.03       0.02
  data required time                                  0.02
  -----------------------------------------------------------
  data required time                                  0.02
  data arrival time                                  -0.18
  -----------------------------------------------------------
  slack (VIOLATED)                                   -0.16


 **********************************************************************
 The Total number of cell counts in the Maximum Timing Path
       U19          1
       U11          1
 *********************************************************************
cell(U19) *1

cell(U11) *1
```



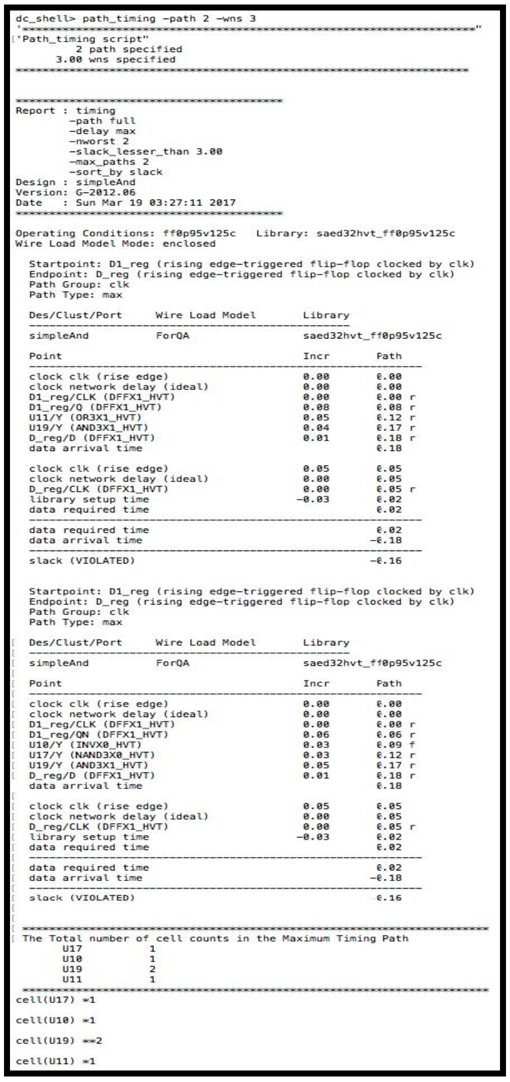
```
dc_shell> path_timing -path 1
'*********************************************************************"
'Path_timing script"
        1 path specified only
*********************************************************************

Information: Updating design information... (UID-85)

****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
        -sort_by slack
Design : simpleAnd
Version: G-2012.06
Date   : Sun Mar 19 02:54:58 2017
****************************************

Operating Conditions: ff0p95v125c   Library: saed32hvt_ff0p95v125c
Wire Load Model Mode: enclosed

  Startpoint: D1_reg (rising edge-triggered flip-flop clocked by clk)
  Endpoint: D_reg (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  simpleAnd          ForQA                 saed32hvt_ff0p95v125c

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  D1_reg/CLK (DFFX1_HVT)                   0.00       0.00 r
  D1_reg/Q (DFFX1_HVT)                     0.08       0.08 r
  U11/Y (OR3X1_HVT)                        0.05       0.12 r
  U19/Y (AND3X1_HVT)                       0.04       0.17 r
  D_reg/D (DFFX1_HVT)                      0.01       0.18 r
  data arrival time                                   0.18

  clock clk (rise edge)                    0.05       0.05
  clock network delay (ideal)              0.00       0.05
  D_reg/CLK (DFFX1_HVT)                    0.00       0.05 r
  library setup time                      -0.03       0.02
  data required time                                  0.02
  -----------------------------------------------------------
  data required time                                  0.02
  data arrival time                                  -0.18
  -----------------------------------------------------------
  slack (VIOLATED)                                   -0.16


 **********************************************************************
 The Total number of cell counts in the Maximum Timing Path
       U19          1
       U11          1
 **********************************************************************
cell(U19) *1

cell(U11) *1
```

Figure 6: Result when wns=10 on the right and when path=1 are specified separately



```
 ************************************************************************
 The Total number of cell counts in the Maximum Timing Path
       U16           2
       U12           8
       U17           2
       U13           2
       U18           4
       U11           2
       U15           2
       U19           8
 ************************************************************************
cell(U16) **2

cell(U12) ********8

cell(U17) **2

cell(U13) **2

cell(U18) ****4

cell(U11) **2

cell(U15) **2

cell(U19) ********8
```

Figure 7 : Example of a simple histogram for a given number of cell counts

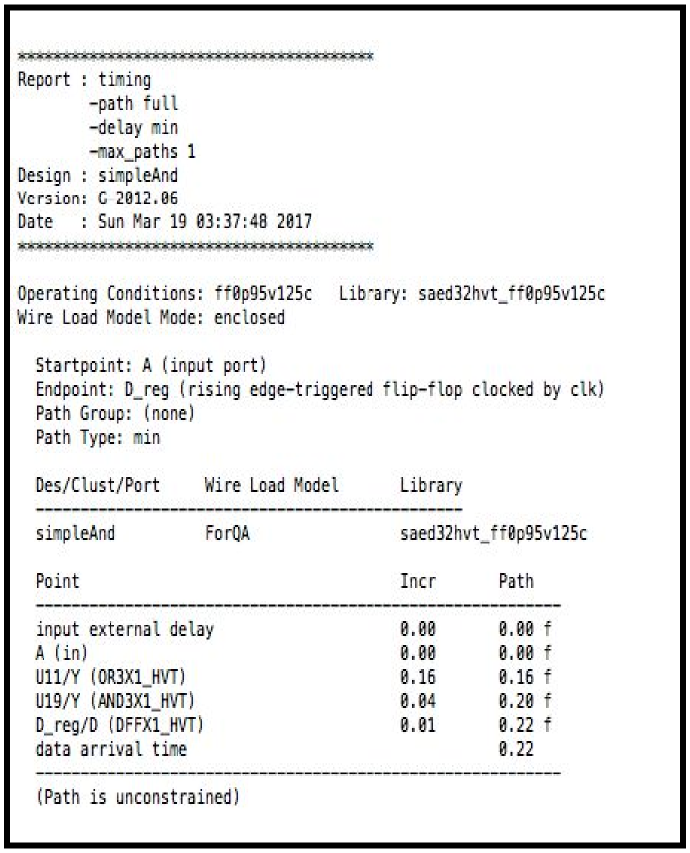


```
dc_shell> path_timing -path 2 -wns 3
'**********************************************************************"
'Path_timing script"
        2 path specified
     3.00 wns specified
**********************************************************************


****************************************
Report : timing
        -path full
        -delay max
        -nworst 2
        -slack_lesser_than 3.00
        -max_paths 2
        -sort_by slack
Design : simpleAnd
Version: G-2012.06
Date   : Sun Mar 19 03:27:11 2017
****************************************

Operating Conditions: ff0p95v125c   Library: saed32hvt_ff0p95v125c
Wire Load Model Mode: enclosed

  Startpoint: D1_reg (rising edge-triggered flip-flop clocked by clk)
  Endpoint: D_reg (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  simpleAnd          ForQA                 saed32hvt_ff0p95v125c

  Point                                    Incr       Path
  ----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  D1_reg/CLK (DFFX1_HVT)                   0.00       0.00 r
  D1_reg/Q (DFFX1_HVT)                     0.08       0.08 r
  U11/Y (OR3X1_HVT)                        0.05       0.12 r
  U19/Y (AND3X1_HVT)                       0.04       0.17 r
  D_reg/D (DFFX1_HVT)                      0.01       0.18 r
  data arrival time                                   0.18

  clock clk (rise edge)                    0.05       0.05
  clock network delay (ideal)              0.00       0.05
  D_reg/CLK (DFFX1_HVT)                    0.00       0.05 r
  library setup time                      -0.03       0.02
  data required time                                  0.02
  ----------------------------------------------------------
  data required time                                  0.02
  data arrival time                                  -0.18
  ----------------------------------------------------------
  slack (VIOLATED)                                   -0.16


  Startpoint: D1_reg (rising edge-triggered flip-flop clocked by clk)
  Endpoint: D_reg (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  simpleAnd          ForQA                 saed32hvt_ff0p95v125c

  Point                                    Incr       Path
  ----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  D1_reg/CLK (DFFX1_HVT)                   0.00       0.00 r
  D1_reg/QN (DFFX1_HVT)                    0.06       0.06 r
  U10/Y (INVX0_HVT)                        0.03       0.09 f
  U17/Y (NAND3X0_HVT)                      0.03       0.12 r
  U19/Y (AND3X1_HVT)                       0.05       0.17 r
  D_reg/D (DFFX1_HVT)                      0.01       0.18 r
  data arrival time                                   0.18

  clock clk (rise edge)                    0.05       0.05
  clock network delay (ideal)              0.00       0.05
  D_reg/CLK (DFFX1_HVT)                    0.00       0.05 r
  library setup time                      -0.03       0.02
  data required time                                  0.02
  ----------------------------------------------------------
  data required time                                  0.02
  data arrival time                                  -0.18
  ----------------------------------------------------------
  slack (VIOLATED)                                   -0.16



 ************************************************************************
 The Total number of cell counts in the Maximum Timing Path
       U17           1
       U10           1
       U19           2
       U11           1
 ************************************************************************
cell(U17) *1

cell(U10) *1

cell(U19) **2

cell(U11) *1
```

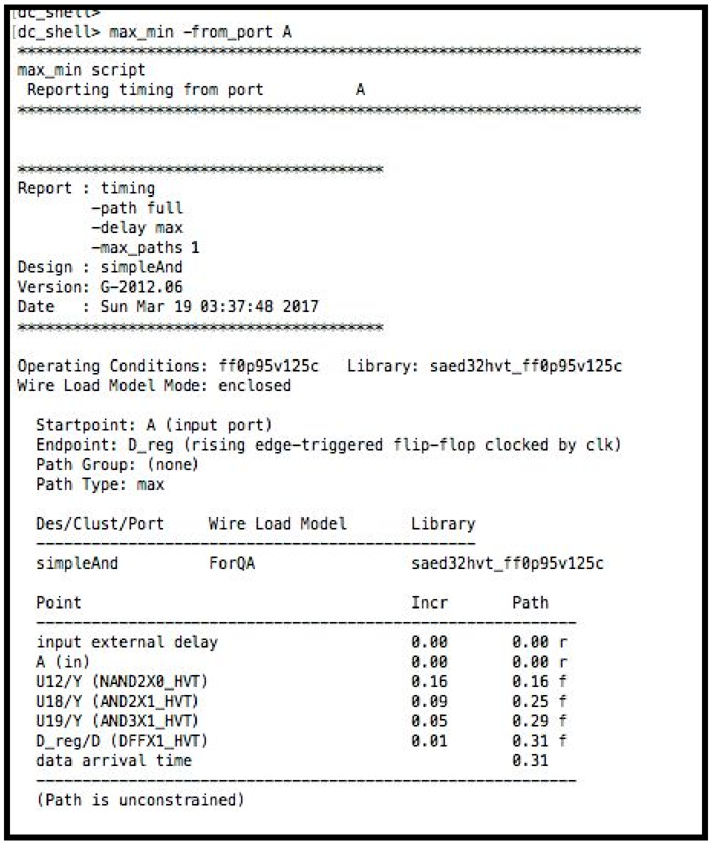
Figure 8: Showing the results when both path and wns are specified

B) **SCRIPT II OUTPUTS:**

There are four possible user inputs for script II. Based on the input, the maximum and minimum timings for that input along with maximum and minimum timing for all the sequential cells on the other side of the sequential is generated. In other words, each input generates four sets of output for which gives max and min information. For example Figure 9 shows the minimum and maximum timing for port A, and figure 10 shows the maximum and minimum timing for sequentials on the other side for the same input *max_min -from U11*

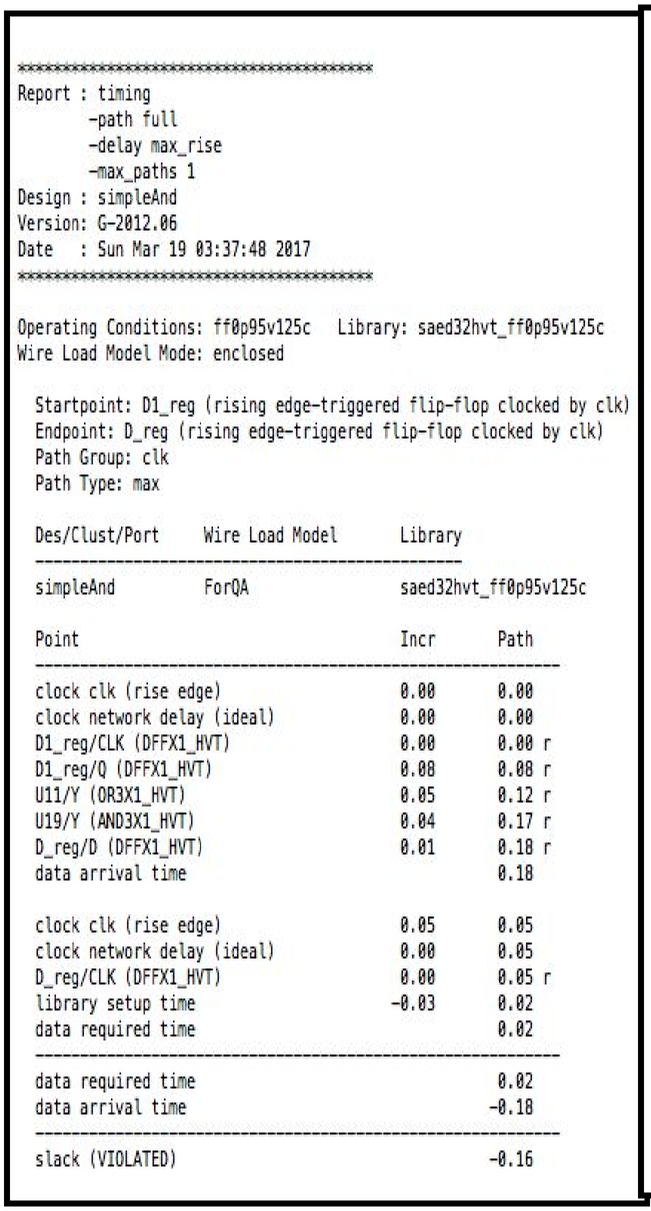


```
****************************************
Report : timing
        -path full
        -delay min
        -max_paths 1
Design : simpleAnd
Version: G 2012.06
Date   : Sun Mar 19 03:37:48 2017
****************************************

Operating Conditions: ff0p95v125c   Library: saed32hvt_ff0p95v125c
Wire Load Model Mode: enclosed

  Startpoint: A (input port)
  Endpoint: D_reg (rising edge-triggered flip-flop clocked by clk)
  Path Group: (none)
  Path Type: min

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  simpleAnd          ForQA                 saed32hvt_ff0p95v125c

  Point                                    Incr       Path
  ---------------------------------------------------------------
  input external delay                     0.00       0.00 f
  A (in)                                   0.00       0.00 f
  U11/Y (OR3X1_HVT)                        0.16       0.16 f
  U19/Y (AND3X1_HVT)                       0.04       0.20 f
  D_reg/D (DFFX1_HVT)                      0.01       0.22 f
  data arrival time                                   0.22
  ---------------------------------------------------------------
  (Path is unconstrained)
```

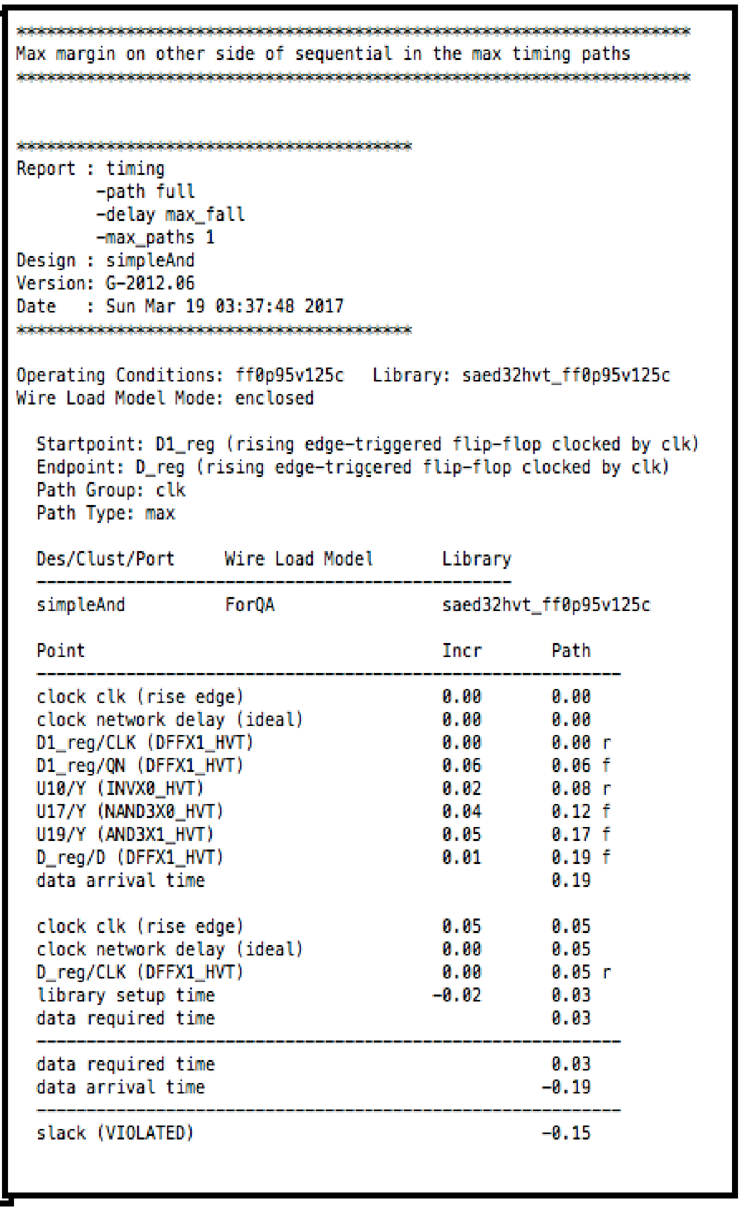


```
[dc_shell>
[dc_shell> max_min -from_port A
**********************************************************************
max_min script
 Reporting timing from port          A
**********************************************************************


****************************************
Report : timing
        -path full
        -delay max
        -max_paths 1
Design : simpleAnd
Version: G-2012.06
Date   : Sun Mar 19 03:37:48 2017
****************************************

Operating Conditions: ff0p95v125c   Library: saed32hvt_ff0p95v125c
Wire Load Model Mode: enclosed

  Startpoint: A (input port)
  Endpoint: D_reg (rising edge-triggered flip-flop clocked by clk)
  Path Group: (none)
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  simpleAnd          ForQA                 saed32hvt_ff0p95v125c

  Point                                    Incr       Path
  ---------------------------------------------------------------
  input external delay                     0.00       0.00 r
  A (in)                                   0.00       0.00 r
  U12/Y (NAND2X0_HVT)                      0.16       0.16 f
  U18/Y (AND2X1_HVT)                       0.09       0.25 f
  U19/Y (AND3X1_HVT)                       0.05       0.29 f
  D_reg/D (DFFX1_HVT)                      0.01       0.31 f
  data arrival time                                   0.31
  ---------------------------------------------------------------
  (Path is unconstrained)
```

Figure 9: Output of the script when report timing for port A is specified



```
****************************************
Report : timing
        -path full
        -delay max_rise
        -max_paths 1
Design : simpleAnd
Version: G-2012.06
Date   : Sun Mar 19 03:37:48 2017
****************************************

Operating Conditions: ff0p95v125c   Library: saed32hvt_ff0p95v125c
Wire Load Model Mode: enclosed

  Startpoint: D1_reg (rising edge-triggered flip-flop clocked by clk)
  Endpoint: D_reg (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  simpleAnd          ForQA                 saed32hvt_ff0p95v125c

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  D1_reg/CLK (DFFX1_HVT)                   0.00       0.00 r
  D1_reg/Q (DFFX1_HVT)                     0.08       0.08 r
  U11/Y (OR3X1_HVT)                        0.05       0.12 r
  U19/Y (AND3X1_HVT)                       0.04       0.17 r
  D_reg/D (DFFX1_HVT)                      0.01       0.18 r
  data arrival time                                   0.18

  clock clk (rise edge)                    0.05       0.05
  clock network delay (ideal)              0.00       0.05
  D_reg/CLK (DFFX1_HVT)                    0.00       0.05 r
  library setup time                      -0.03       0.02
  data required time                                  0.02
  -----------------------------------------------------------
  data required time                                  0.02
  data arrival time                                  -0.18
  -----------------------------------------------------------
  slack (VIOLATED)                                   -0.16
```



```
***************************************************************************
Max margin on other side of sequential in the max timing paths
***************************************************************************


********************************************
Report : timing
        -path full
        -delay max_fall
        -max_paths 1
Design : simpleAnd
Version: G-2012.06
Date   : Sun Mar 19 03:37:48 2017
********************************************

Operating Conditions: ff0p95v125c   Library: saed32hvt_ff0p95v125c
Wire Load Model Mode: enclosed

  Startpoint: D1_reg (rising edge-triggered flip-flop clocked by clk)
  Endpoint: D_reg (rising edge-triggered flip-flop clocked by clk)
  Path Group: clk
  Path Type: max

  Des/Clust/Port     Wire Load Model       Library
  ------------------------------------------------
  simpleAnd          ForQA                 saed32hvt_ff0p95v125c

  Point                                    Incr       Path
  -----------------------------------------------------------
  clock clk (rise edge)                    0.00       0.00
  clock network delay (ideal)              0.00       0.00
  D1_reg/CLK (DFFX1_HVT)                   0.00       0.00 r
  D1_reg/QN (DFFX1_HVT)                    0.06       0.06 f
  U10/Y (INVX0_HVT)                        0.02       0.08 r
  U17/Y (NAND3X0_HVT)                      0.04       0.12 f
  U19/Y (AND3X1_HVT)                       0.05       0.17 f
  D_reg/D (DFFX1_HVT)                      0.01       0.19 f
  data arrival time                                   0.19

  clock clk (rise edge)                    0.05       0.05
  clock network delay (ideal)              0.00       0.05
  D_reg/CLK (DFFX1_HVT)                    0.00       0.05 r
  library setup time                      -0.02       0.03
  data required time                                  0.03
  -----------------------------------------------------------
  data required time                                  0.03
  data arrival time                                  -0.19
  -----------------------------------------------------------
  slack (VIOLATED)                                   -0.15
```

Figure 10 : Showing the max and min margin on the other side of the sequential

V. Conclusions

In short , the two specifications were successfully implemented in two scripts. The script 1 , generated the total count of cells in the maximum timing path, the cell delays and the creation of histogram for the total cell count. A histogram with cell delays has not been implemented. Similarly, script 2 generated timing margin output for given cells, nets or I/O ports. Our scripts are generic and are not design dependent. As the part of the verification, we modeled a simple design circuit to test and verify the correctness of all outputs generated by our script. Both script

1 and script 2 have also implemented and verified successfully. Starting from the scratch, and learning to code in tcl so as to create and run the script in Synopsys' design tools, we had a profound opportunity to learn and experience the design automation challenges in the real world like design environment.

VI. REFERENCES

[1] Using Tcl With Synopsys® Tools Version M-2016.12, December 2016
[2] IC Compiler™ Variables and Attributes Version M-2016.12, December 2016
[3] Simulation and Synthesis Techniques for Asynchronous FIFO Design Clifford E. Cummings, Sunburst Design, Inc. cliffc@sunburst-design.com
[4] Tcl wiki page (http://wiki.tcl.tk/)

*Manuscript submitted on **March 19, 2017** by **Aanchal Chobisa, Bhaskar Venkatramaiah, Saroj Bardewa** who are graduate students in the ECE Department of Portland State University, Portland, OR. This paper is submitted as part of ASIC: Modeling and Synthesis (ECE 581) course* during the Winter Term of 2017.

Appendix-A

Script I

```
# Title:       path_timing.tcl
# Description: This Tcl procedure generates specific outputs given user input.
# INPUTS:
# The following are the inputs to the module:
# N   -  Top N paths, where N is the number of paths
# WNS_val -   WNS value
# OUTPUTS:
# Total_cells  - Total number of cells
# N_cells      - Number of times each cell found
# delay_cell   - Delay of each cell
# Options:     -N            Number of top N paths
#              -WNS_val      Paths with the specified WNS
#              -N -WNS_Val   Number of cells with WNS
#Usage:        prompt> source path_timing.tcl
#Authors:      Saroj, Bhaskar, Aachal
proc path_timing args {
   suppress_message UID-101
   ################################################
   # Parse the user inputs
   set option [lindex $args 0]
   set value1 [lindex $args 1]
   set option2 [lindex $args 2]
   set value2 [lindex $args 3]
   ###################################################
   if {[string match -help* $option]} {
       echo " "
       echo "This tcl script finds the top N paths with or"
       echo "without WNS specified and draws the histogram"
       echo "to show the number of times cells appeared in"
       echo "those paths"
```

```
        echo "-path : report top N paths"
        echo "-wns  : report worst negative paths"
        return
        } elseif {[string match -path* $option] && ([string match -wns* 
$option2]==0)} {
           # Only number of paths specified
       set Npath [expr $value1*1]
       echo 
"**********************************************************************"
           echo "Path_timing script"
           echo [format "%10s path specified only" $Npath]
           echo 
"**********************************************************************"
           echo ""
           # Perform the operation
           report_timing -path full -delay max -nworst $Npath -sort_by slack
     set cells 0
     set cellFound 0
     array set count_cell {}
     set all_paths [get_timing_paths -path full -delay max -nworst $Npath]
     # To find the Total Number of cells in the Max Timing Path
         foreach_in_collection path $all_paths {
         set  start [get_attribute $path startpoint]
                 set  path_start [get_attribute $start full_name]
         #echo sizeof_collection [get_attribute $path startpoint]
         set end [get_attribute $path endpoint]
         set path_end [get_attribute $end full_name]
         #echo "-------------------------------------------------------------"
         #puts "**Path between : $path_start - $path_end**"
             set all_points [get_attribute $path points]
         foreach_in_collection point $all_points {
             set start [get_attribute $point object]
             set name [get_attribute $start full_name]
```

```
            set cell_name [get_cells -of_object "$name"]
            set cell [get_attribute $cell_name full_name]
            # To get the combinational cells in the Timing path
            set t_cell [get_attribute $cell is_combinational]
            #only if the gate is a combinational cell
            if {$t_cell=="true"} {
                if {[array size count_cell]==0} {
                 set count_cell($cell) 1
                 set count 1
                 #echo [format "%10s %10s" $cell $count ]
                } else { foreach { name_cell count} [array get count_cell] {
                             if {[string match $name_cell $cell]} {
                         set newcount [expr 1 + $count]
                         #set newcount [expr $ewcount/2]
                         set count_cell($cell) $newcount
                         set cellFound 1
                        # echo [format "%10s %10s" $name_cell $count]
                         break
                        # echo [format "%10s %10s" $name_cell $newcount]
                    } else { set cellFound 0 }}
                    if {$cellFound==0} { set count_cell($cell) 1}
                    }}}}
            echo "
**********************************************************************  "
            echo " The Total number of cell counts in the Maximum Timing Path "
            foreach { name_cell count} [array get count_cell] { echo [format
"%10s %10s" $name_cell [expr $count/2]] }
            echo "
**********************************************************************  "
#array set arr $count_cell
foreach element [array names count_cell] {
    puts -nonewline "cell($element) "
    for {set i 0} {$i <= [expr $count_cell($element)/2 - 1]} {incr i} {
```

```
    #echo $count_cell($element)
        puts -nonewline "*"
    }
    puts -nonewline "[expr $count_cell($element)/2]"
    #echo $count_cell($element)
    puts "\n"
}  return
      } elseif {[string match -path* $option]&& ([string match -wns* 
$option2])} {
          # Both number of path and wns specified
      set Npath [expr $value1*1]
          set wns [expr $value2*1.0]
      echo 
"**********************************************************************"
          echo "Path_timing script"
          echo [format "%10s path specified" $Npath]
          echo [format "%10.2f wns specified" $wns]
          echo 
"**********************************************************************"
          echo ""
          # Perform the operation
          report_timing -path full -delay max -nworst $Npath -slack_lesser_than 
$wns -sort_by slack
            # Operation performed to find the Tital number of cells in the 
Maximum timing path
            set cells 0
            set cellFound 0
            array set count_cell {}
            set all_paths [get_timing_paths -path full -delay max -nworst 
$Npath -slack_lesser_than $wns ]
            foreach_in_collection path $all_paths {
            set  start [get_attribute $path startpoint]
                set  path_start [get_attribute $start full_name]
        #echo sizeof_collection [get_attribute $path startpoint]
        set end [get_attribute $path endpoint]
        set path_end [get_attribute $end full_name]
        #echo "-------------------------------------------------------------"
        #puts "**Path between : $path_start - $path_end**"
            set all_points [get_attribute $path points]
        foreach_in_collection point $all_points {
            set start [get_attribute $point object]
            set name [get_attribute $start full_name]
            set cell_name [get_cells -of_object "$name"]
            set cell [get_attribute $cell_name full_name]
            #echo [format "%10s" $cell]
            set t_cell [get_attribute $cell is_combinational]
            #only if the gate is a combinational cell
            if {$t_cell=="true"} {
                if {[array size count_cell]==0} {
                 set count_cell($cell) 1
                 set count 1
                 #echo [format "%10s %10s" $cell $count ]
                } else { foreach { name_cell count} [array get count_cell] {
                         if {[string match $name_cell $cell]} {
                         set newcount [expr 1 + $count]
                         #set newcount [expr $ewcount/2]
                         set count_cell($cell) $newcount
                         set cellFound 1
                        # echo [format "%10s %10s" $name_cell $count]
                         break
                        # echo [format "%10s %10s" $name_cell $newcount]
                    } else { set cellFound 0 }}
                    if {$cellFound==0} { set count_cell($cell) 1}
                        }}}}
            echo "
**********************************************************************  "
```

```
            echo " The Total number of cell counts in the Maximum Timing Path "
            foreach { name_cell count} [array get count_cell] { echo [format
"%10s %10s" $name_cell [expr $count/2]] }
            echo "
*********************************************************************** "
            #array set arr $count_cell
foreach element [array names count_cell] {
    puts -nonewline "cell($element) "
    for {set i 0} {$i <= [expr $count_cell($element)/2 - 1]} {incr i} {
    #echo $count_cell($element)
        puts -nonewline "*"
    }
    puts -nonewline "[expr $count_cell($element)/2]"
    #echo $count_cell($element)
    puts "\n"
}       return
       } elseif {[string match -wns* $option]} {
          #Only WNS specified
          set wns [expr $value1*1.0]
          echo
"**********************************************************************"
          echo "Path_timing script"
          echo [format "%10.2f wns only was specified" $wns]
          echo
"**********************************************************************"
          # Perform the operation
      report_timing -path full -delay max -sort_by slack -slack_lesser_than
$wns
      set cells 0
    set cellFound 0
    array set count_cell {}
    set all_paths [get_timing_paths -path full -delay max  -slack_lesser_than
$wns]
```

```
foreach_in_collection path $all_paths {
set  start [get_attribute $path startpoint]
        set  path_start [get_attribute $start full_name]
#echo sizeof_collection [get_attribute $path startpoint]
set end [get_attribute $path endpoint]
set path_end [get_attribute $end full_name]
#echo "-------------------------------------------------------------"
#puts "**Path between : $path_start - $path_end**"
    set all_points [get_attribute $path points]
foreach_in_collection point $all_points {
    set start [get_attribute $point object]
    set name [get_attribute $start full_name]
    set cell_name [get_cells -of_object "$name"]
    set cell [get_attribute $cell_name full_name]
    #echo [format "%10s" $cell]
    set t_cell [get_attribute $cell is_combinational]
    #only if the gate is a combinational cell
    if {$t_cell=="true"} {
        if {[array size count_cell]==0} {
         set count_cell($cell) 1
         set count 1
         #echo [format "%10s %10s" $cell $count ]
        } else { foreach { name_cell count} [array get count_cell] {
                     if {[string match $name_cell $cell]} {
                 set newcount [expr 1 + $count]
                 #set newcount [expr $ewcount/2]
                 set count_cell($cell) $newcount
                 set cellFound 1
                # echo [format "%10s %10s" $name_cell $count]
                 break
                # echo [format "%10s %10s" $name_cell $newcount]
            } else { set cellFound 0 }}
            if {$cellFound==0} { set count_cell($cell) 1}
```

```
                    }}}}
            echo "
********************************************************************** "
            echo " The Total number of cell counts in the Maximum Timing Path "
            foreach { name_cell count} [array get count_cell] { echo [format
"%10s %10s" $name_cell [expr $count/2]] }
            echo "
********************************************************************* "
            #array set arr $count_cell
foreach element [array names count_cell] {
    puts -nonewline "cell($element) "
    for {set i 0} {$i <= [expr $count_cell($element)/2 - 1]} {incr i} {
    #echo $count_cell($element)
        puts -nonewline "*"
    }
    puts -nonewline "[expr $count_cell($element)/2]"
    #echo $count_cell($element)
    puts "\n"
}   return
    } else {
          echo " "
          echo "  Message: Option Required"
          echo "  Usage:  path_timing \[-path\] \[-wns\]"
          echo " "
          return
      }
}
  ########################################################
 define_proc_attributes path_timing \
   -info "Procedure to report top N paths in the design" \
   -define_args {
   {-path "report top N paths"}
   {-wns  "report worst negative paths"}}
```

Appendix-B

Script II

```
# Title:          max_min.tcl
# Description:
# This Tcl procedure generates specific outputs given user input for maximum
# paths.
# INPUTS:
# The following are the inputs to the module:
# from_port    - Report timing from a port
# to_port      - Report timing to a port
# nets         - Report timing for all nets
# cells        - Report timing for all cells
# OUTPUTS:
# from_port        - max min timing from a port
# to_port          - max min timing to a port
# nets             - max min timing for nets
# cells            - max min timing for cells
#
# Options:        -from_port       Specify input port
#                 -to_port         Specify output port
#                 -nets            Specify nets
#                 -cells           Specify cells
#
# Usage:          prompt> max_min \[-from_port {A}]\[-to_port
{A}]\[-cells\]\[-nets\]
# Authors:        Saroj Bardewa,Bhaskar Venkataramaiah, Aanchal Chobisa
proc max_min args {
   suppress_message UID-101
   ################################################
   # Parse the user inputs
   set com_args0 [lindex $args 0]
   set com_args1 [lindex $args 1]
```

```
    # Default maximum timing paths to report
    set Npath 100
    ###################################################
    if {[string match -help* $com_args0]} {
        echo " "
        echo "This tcl script reports maximun and minimum"
        echo "timing for cells/nets/ an IO port in  "
        echo "maximum timing paths"
        echo "-from_port  : report max and min timing from a port"
        echo "-to_port    : report max and min timing to a port "
        echo "-cells      : report min and max timing margin of all cells"
        echo "-nets       : report min and max timing of all nets"

        } elseif {[string match -cells* $com_args0]} {
           # Cell information
       echo
"*********************************************************************"
           echo "max_min script"
       echo "Min and max margin for each cells in the max timing paths"
           echo
"*********************************************************************"
           echo ""
           # Perform the operation
           report_timing -path full -delay max -nworst $Npath
           report_timing -path full -delay min -nworst $Npath

       echo
"*********************************************************************"
       echo "Max margin on other side of sequential in the max timing paths"
           echo
"*********************************************************************"
           echo ""
       report_timing -delay max_fall -nworst $Npath
```

```
      report_timing -delay max_rise -nworst $Npath
       } elseif {[string match -nets* $com_args0]} {
      # Nets in the maximum timing path
      echo
"*********************************************************************"
          echo "max_min script"
          echo "Reporting timing margin of nets"
          echo
"*********************************************************************"
          echo ""
          # Perform the operation
          report_timing -nets -delay max -nworst $Npath
      report_timing -nets -delay min -nworst $Npath

      echo
"*********************************************************************"
      echo "Max margin on the other side of sequential in the max timing paths"
          echo
"*********************************************************************"
          echo ""
      report_timing -delay max_fall -nworst $Npath
      report_timing -delay max_rise -nworst $Npath

       } elseif {[string match -from_port* $com_args0]} {
          # Both number of path and wns specified
      set port $com_args1
      echo
"*********************************************************************"
          echo "max_min script"
          echo [format " Reporting timing from port %10s " $port]
          echo
"*********************************************************************"
          echo ""
      report_timing -from $port -delay max -nworst $Npath
      report_timing -from $port -delay min -nworst $Npath

      echo
"********************************************************************"
      echo "Max margin on other side of sequential in the max timing paths"
          echo
"********************************************************************"
          echo ""
      report_timing -delay max_fall -nworst $Npath
      report_timing -delay max_rise -nworst $Npath
       } elseif {[string match -to_port* $com_args0]} {
          # Both number of path and wns specified
      set port $com_args1
      echo
"********************************************************************"
          echo "max_min script"
          echo [format " Reporting timing to port %10s " $port]
          echo
"********************************************************************"
          echo ""
      report_timing -to $port -delay max -nworst $Npath
      report_timing -to $port -delay min -nworst $Npath

      echo
"********************************************************************"
      echo "Max margin on other side of sequential in the max timing paths"
          echo
"********************************************************************"
          echo ""
      report_timing -delay max_fall -nworst $Npath
      report_timing -delay max_rise -nworst $Npath
       } else {
```

```
       echo " "
           echo "  Message: Option Required"
           echo "  Usage  : max_min\[-from_port {A}]\[-to_port
{A}]\[-cells\]\[-nets\]"
           echo " "
        }
}

   ########################################################
   define_proc_attributes max_min \
   -info "Procedure to report timing margins in the design" \
   -define_args {
   {-from_port  "report max and min timing from a port "}
   {-to_port    "report max and min timing to a port "}
   {-cells      "report min and max timing margin of all cells"}
   {-nets       "report min and max timing of all nets"}}
```